Noise Schedules for Diffusion Probabilistic Models

Nikolas Kirschstein, University of Oxford

Abstract

Diffusion Probabilistic Models (DPM) gained attention recently due to their mathematical properties and competitive performance in image generation tasks. A major ingredient in DPMs is the noise schedule which dictates the noising process that a denoising model must learn to undo. We explore and compare different choices for the noise schedule in a common framework on a non-standard dataset.

1 Introduction

With DALL-E 2 [10], Imagen [13], and Stable Diffusion [11] in 2022 alone, diffusion probabilistic models have been on the rise. They were first introduced by Sohl-Dickstein et al. [15] and base off a diffusion process where a small amount of noise is repeatedly added to a sample from the data distribution until it becomes indistinguishable from random samples. The model then must learn to undo this process, so that given any random noise as input it can "denoise" it into a sample that follows the data distribution. Ho et al. [3] were the first to achieve competitive image generation results with DPMs, and Dhariwal et al. [1] later showed that they are even capable of outperforming state-of-the-art GAN-based models.

The performance of DPMs heavily relies on the choice of the noise schedule which governs the amount of noise added in each diffusion step. Ho et al. [3] originally used a linear schedule due to its simplicity but they noted that this is an arbitrary choice and more research on the noise schedule is necessary. Nichol et al. [8] employ a cosine-based schedule and find that, while it does not improve over the linear schedule per se, it might make the model more robust to fast sampling in the denoising process where multiple steps are combined. Kigma et al. [6] propose to simply let a small neural network learn the schedule jointly with the main model. In this vein, they indeed improve on the log-likelihood but not on other important metrics that capture generation quality. Since the results are inconclusive and the linear schedule is the most straightforward to implement, researchers seem to have settled with it, focusing on other fundamental aspects like guidance [1, 4] and sampling efficiency [16, 18, 14].

Contributions We are not aware of any comparative study that isolates the noise schedule of DPMs and addresses it in-depth. In this work, we provide an extensive analytical and empirical comparison of noise schedules and also include a constant schedule in the discussion which, to the best of our knowledge, is not covered in the relevant literature. Furthermore, we base our experiments off the Oxford Flowers dataset [9] to obtain models and results that are independent of benchmark datasets like CIFAR-10 and ImageNet which are almost exclusively used in related work. Our codebase will be made publicly available as open source.

2 Diffusion Probabilistic Models

In this section, we provide a formal description of DPMs. Since they are probabilistic models, we view the input as a standardised random variable $\mathbf{x}_0 \in \mathbb{R}^d$ with $\mathbb{E}[\mathbf{x}_0] = 0$ and $\mathrm{Var}[\mathbf{x}_0] = 1$ originating from some underlying distribution $q(\mathbf{x}_0)$. We model the diffusion steps as a sequence of T *latent variables* $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^d$ of the same dimensionality. Note that recent advances loosen this last requirement and resort to a lower-dimensional latent space for the sake of efficiency [11].

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2.1 Forward Process

The diffusion process itself in DPMs is constructed as a non-homogeneous Markov chain over the latent variables [15, 3]. Hence, the posterior PDF $q(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid \mathbf{x}_0)$ factorises as follows:

$$q(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \underbrace{\mathbf{x}_{t-2} \ldots, \mathbf{x}_0}_{\text{independent of } \mathbf{x}_t}) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \tag{1}$$

A diffusion step is modelled by drawing from a normal distribution centred around a slightly down-scaled version of the previous latent variable. The noising intensity of each step is governed by our object of interest, the *noise schedule* $(\alpha_t)_{t=1}^T \in (0,1)^T$, discussed in detail in Section 3. We have:

$$\begin{aligned} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) &\coloneqq \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}) \\ \text{i.e., } \mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon} \text{ for } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \end{aligned} \tag{2}$$

As we will see, the cumulative products $\bar{\alpha}_t \coloneqq \prod_{s=1}^{t} \alpha_s$ deserve a shorthand. By recursively unfolding the recurrence (2) and exploiting the way Gaussians combine into Gaussians, we obtain an expression for the marginal distribution of $\mathbf{x}_t$ in terms of only the initial input $\mathbf{x}_0$.

Lemma 1 (Instant Sampling)**.** *Sampling $\mathbf{x}_t$ can be done in closed form:*

$$\begin{aligned} q(\mathbf{x}_t \mid \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}) \\ \textit{i.e., } \mathbf{x}_t &= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \textit{ for } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \end{aligned} \tag{3}$$

Equation (3) tells us that, to obtain a latent variable realisation, we do not need to pass through all intermediate steps but can fast-forward to any desired noise level directly. Furthermore, it yields an asymptotic statement about the end of the chain.

Corollary 1. *As $T \to \infty$, the latent $\mathbf{x}_T$ becomes standard Gaussian, since $\bar{\alpha}_T \xrightarrow{T\to\infty} 0$.*

2.2 Backward Process

To reverse the diffusion process, we need to get hold of $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$. However, obtaining this reverse conditional distribution exactly is not tractable as its computation via Bayes' theorem involves marginalising over the unknown data distribution. Thus, we estimate it using a parameterised model $p_{\boldsymbol{\theta}}(\mathbf{x}_0, \ldots, \mathbf{x}_T)$ of the joint distribution that approximates the transition distributions so that $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \approx q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$. Motivated by $q(\mathbf{x}_T \mid \mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \boldsymbol{I})$ for T large enough according to Corollary 1, we initialise the reverse process by drawing from a standard normal distribution, i.e., set $p_{\boldsymbol{\theta}}(\mathbf{x}_T) \coloneqq \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \boldsymbol{I})$. For the actual $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$, we utilise Bayes' theorem and tedious algebraic manipulations to obtain the corresponding forward process posteriors [7].

Theorem 1. *When conditioning on $\mathbf{x}_0$, the reverse process transitions are given as*

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}, \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\sigma}_t^2 \boldsymbol{I}) \tag{4}$$

$$\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0) \coloneqq \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\mathbf{x}_0 \tag{5}$$

$$\tilde{\sigma}_t^2 \coloneqq \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \tag{6}$$

Thus, to match $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ closely to $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$, we parameterise it as a Gaussian as well. While, interestingly, the variance $\tilde{\sigma}_t^2$ only depends on the schedule and we can hence match it exactly, we inherently do not have access to $\mathbf{x}_0$ and thus to obtain the mean $\tilde{\boldsymbol{\mu}}$ we need to estimate it from the available information. Rearranging (3) in Lemma 1 suggests that we can estimate $\mathbf{x}_0$ using a neural network that predicts $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \approx \boldsymbol{\epsilon}$. Hence, we parameterise the denoising process as

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) &\coloneqq \mathcal{N}(\mathbf{x}_{t-1}; \bar{\boldsymbol{\mu}}(\mathbf{x}_t, \tfrac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}), \tilde{\sigma}_t^2 \boldsymbol{I}) \\ \text{i.e., } \mathbf{x}_{t-1} &= \tfrac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \tfrac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)) + \tilde{\sigma}_t \boldsymbol{\epsilon} \text{ for } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \end{aligned} \tag{7}$$

Note that we could equivalently parameterise $\mathbf{x}_0$ just as well. The architecture for $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ depends on the type of data – for images, CNN auto-encoders like a U-Net are a common choice, since they bring a suitable inductive bias and have empirically performed well on related tasks [12, 3, 1]. The timestep input can be encoded via positional embeddings like in the Transformer architecture [17]. Since our focus is not on architecture, we simply employ the deep U-Net from Ho et al. [3].

2.3 Optimisation Objective

The interaction of $p_{\boldsymbol{\theta}}$ and q is reminiscent of the variational auto-encoder (VAE) paradigm [5]. In fact, we can view a DPM as a hierarchical VAE where the chain is jointly optimised and the encoder is parameter-free. For VAEs, the natural optimisation objective is maximising the log-likelihood of the data. It has a tractable "evidence" lower bound (ELBO) which serves as a suitable proxy.

Lemma 2 (ELBO). *For a hierarchical VAE with encoder $q_{\boldsymbol{\psi}}$ and decoder $p_{\boldsymbol{\theta}}$, we have*

$$\mathbb{E}_q \log(p_{\boldsymbol{\theta}}(\mathbf{x})) \geq \mathbb{E}_q \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_T)}{q_{\boldsymbol{\psi}}(\boldsymbol{x}_1, \dots \boldsymbol{x}_T \mid \mathbf{x}_0)} \right]. \tag{8}$$

With a series of algebraic manipulations leveraging the Markov property analogous to (1) and applying log rules, the general ELBO in (8) decomposes for the case of DPMs into the more interpretable expression [15, 3]

$$\underbrace{\mathbb{E}_q[\log(p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1))]}_{=:\mathcal{L}_0 \text{ (reconstruction quality)}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_q[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))]}_{=:\mathcal{L}_{t-1} \text{ (denoising matching quality)}} - \underbrace{\mathbb{E}_q[D_{\mathrm{KL}}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_T))]}_{=:\mathcal{L}_T \text{ (prior matching quality)}}.$$

Note that the last summand $\mathcal{L}_T$ is a small constant vanishing for $T \to \infty$ due to Corollary 1 and can therefore be ignored for optimisation. Similarly, although not constant, the first summand $\mathcal{L}_0$ is heavily dominated by the summation term, which is thus the only one we consider for optimisation. Since the KL divergence appearing in $\mathcal{L}_{t-1}$ is comparing Gaussian PDFs, it can be calculated exactly. By additionally unpacking $\mathbf{x}_t$ according to Lemma 1, the true reverse transition according to (4), and the estimated transition according to (7), we get

$$\mathcal{L}_{t-1} = \frac{1}{2\bar{\sigma}_t^2} \frac{(1-\alpha_t)^2}{\alpha_t(1-\bar{\alpha}_t)} \mathbb{E}_{\boldsymbol{\epsilon}, \mathbf{x}_t} \left[\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \boldsymbol{\epsilon}\| \right] \overset{(6)}{=} \frac{1}{2} \frac{(1-\alpha_t)}{\alpha_t(1-\bar{\alpha}_{t-1})} \mathbb{E}_{\boldsymbol{\epsilon}, \mathbf{x}_t} \left[\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \boldsymbol{\epsilon}\| \right]. \tag{9}$$

Finally, by expressing the summation term $\sum \mathcal{L}_{t-1}$ as expectation over a discrete uniform distribution and ignoring time-independent constants, we arrive at the following optimisation problem:

$$\boldsymbol{\theta}^* \coloneqq \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\substack{t \sim \mathcal{U}(\{2, \dots, T\}), \\ \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}), \boldsymbol{x}_0 \sim q(\mathbf{x}_0)}} \left[\frac{(1-\alpha_t)}{\alpha_t(1-\bar{\alpha}_{t-1})} \|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) - \boldsymbol{\epsilon}\| \right] \tag{10}$$

In practice, we run mini-batch gradient descent on Monte Carlo estimates of the expectation in (10). This means that for every input in the batch, we sample a time step and Gaussian noise to compute the loss, which is then averaged over the batch dimension.

3 Noise Schedules

Now we turn to the noise schedule $(\alpha_t)_{t=1}^T \in (0,1)^T$ which is the remaining fundamental modelling choice. It determines the sampling procedure in (3) and the optimisation term in (10). We consider both cases of fixed and learned schedules.

3.1 Fixed Schedules

The straightforward approach is to treat the noise schedule as a hyperparameter and thus fix it from the very beginning. We discuss different options and illustrate the resulting latent mean $\bar{\alpha}_t$, which we can interpret as "signal strength" due to Equation (3), for $T = 1000$ in Figure 1.

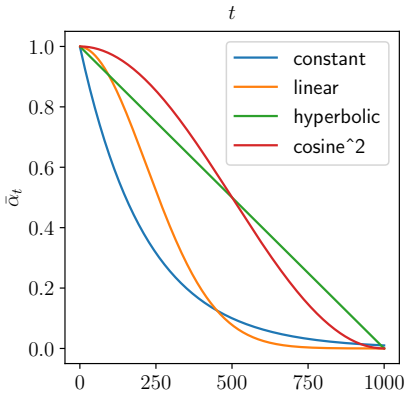
Constant Schedule. The simplest choice is to use a constant variance $1 - \alpha_t = C$ in each step. This leads to an exponential signal decay $\bar{\alpha}_t = (1-C)^t$. Since we need $\bar{\alpha}_T = \varepsilon$ for a small $\varepsilon > 0$, we set $C = 1 - \sqrt[T]{\varepsilon}$. In this work, we choose $\varepsilon = 0.01$ and thus

$$\alpha_t = \sqrt[T]{0.01} \quad \rightsquigarrow \quad \bar{\alpha}_t = 0.01^{t/T}. \tag{11}$$

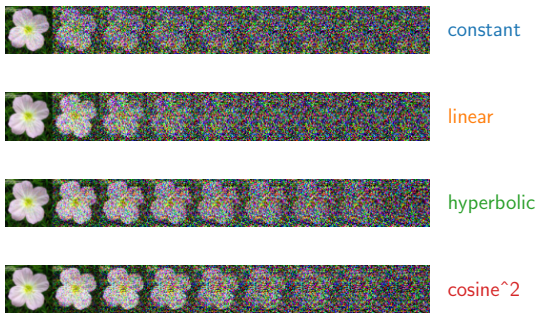
Linear Schedule. Ho et al. [3] apply a schedule where the step-wise noising variance $1 - \alpha_t$ increases linearly from 0.0001 to 0.02, i.e.

$$\alpha_t = 0.9999 - \tfrac{1}{T} 0.0199t. \quad \rightsquigarrow \quad \bar{\alpha}_t = \textstyle\prod_{s=1}^{t} \alpha_s \tag{12}$$

Figure 1a shows that $\bar{\alpha}_t$ has a long final tail of quasi-zero signal. This is sub-optimal as the last 20% of the time steps are essentially not contributing to the diffusion process, and Nichol et al. show they can indeed be skipped without forfeiting performance [8].



(a) Signal strength over time.



(b) Example input after every $\frac{T}{8}$ noising steps.

Figure 1: Comparison fixed noise schedules for $T = 1000$. We clearly see the exponential noise explosion of the constant schedule, the long all-noise tail of the linear schedule, the uniform noising behaviour of the hyperbolic schedule, and the warm-up/cool-down effect of the cosine^2 schedule.

Hyperbolic Schedule. In the original paper, Sohl-Dickstein et al. [15] draw from the idea that the forward process should remove $\frac{1}{T}$ of the original signal in each step. That is, the first step destroys $\frac{1}{T}$ of the input, the second step destroys $\frac{1}{T-1}$ of the first latent variable (which has $\frac{T-1}{T}$ signal left) etc. These fractions determine the step variance $1 - \alpha_t = \frac{1}{T-t+1}$ and thus we get

$$\alpha_t \coloneqq \frac{1}{1 + \frac{1}{T-t}} \quad \rightsquigarrow \quad \bar{\alpha}_t = \prod_{s=1}^{t} \frac{T-s}{T-(s-1)} = 1 - \tfrac{t}{T}. \tag{13}$$

Hence by construction, the signal strength $\bar{\alpha}_t$ decays linearly. In the implementation, we force α_t to stay within the interval $[10^{-4}, \frac{1}{2}]$ for numerical stability.

Cosine^2 Schedule. Nichol et al. [8] dislike the linear schedule's long flat $\bar{\alpha}_t$ tail and use a squared cosinusoid $f(t) = \cos(\frac{\pi}{2}\frac{t/T+s}{1+s})^2$ due to its symmetry and quasi-linear drop midway through:

$$\alpha_t \coloneqq \frac{f(t)}{f(t-1)} \quad \rightsquigarrow \quad \bar{\alpha}_t = \frac{f(t)}{f(0)} \tag{14}$$

The authors introduce a small shift $s = 0.008$ as without it the noise in the initial time steps is imperceptible due to the discrete data, which slows down training considerably. Furthermore, they cap α_t at 0.999 for numerical stability.

3.2 Learned Schedule

Instead of fixing the noise schedule beforehand, Kigma et al. [6] propose to parameterise it and train it jointly with the model. To this end, they introduce a monotonic network

$$\gamma_{\boldsymbol{\eta}}(t) \coloneqq l_1(t/T) + l_3\Big(2 * \sigma\Big(l_2\big(2(t/T - 0.5)\big) - 0.5\Big)\Big)/n \tag{15}$$

where σ is the expit sigmoid activation and $l_1 : \mathbb{R} \to \mathbb{R}, l_2 : \mathbb{R} \to \mathbb{R}^n, l_3 : \mathbb{R}^n \to \mathbb{R}$ are monotonic linear layers, i.e., their parameters are restricted to be non-negative. The noise schedule can then be defined as

$$\alpha_t \coloneqq \frac{\sigma(-\gamma_{\boldsymbol{\eta}}(t))}{\sigma(-\gamma_{\boldsymbol{\eta}}(t-1))} \quad \rightsquigarrow \quad \bar{\alpha}_t = \sigma(-\gamma_{\boldsymbol{\eta}}(t)). \tag{16}$$

The key difficulty with the learned approach is parameter initialisation. Since the optimisation procedure samples using the schedule from the very first instance, we must ensure that $\bar{\alpha}_t$ has a reasonable initial shape and is numerically stable. The highly specific form of $\gamma_{\boldsymbol{\eta}}$ in (15) stems from this concern. If we initialise the parameters of l_2 and l_3 according to a standard normal distribution, then the rescalings and shifts guarantee that the second summand overall is standard Gaussian, too. If we furthermore initialise $l_1(t) = l_{\min} + (l_{\max} - l_{\min})t$ with constants $l_{\min}, l_{\max} \geq 0$, then any initialisation of the network will yield $l_1(t/T)$ slightly distorted by standard Gaussian fluctuations. In this work, we set $l_{\min} = -10$ and $l_{\max} = 10$.

4 Experiments

To empirically compare the noise schedules, we train DPMs on the Oxford Flowers dataset [9] from the Visual Geometry Group. It is a comparatively small set of around 8000 photos of common UK flowers and thus well-suited for an unconditional image generation task. We resize and centre-crop all photos to 64x64 pixels and augment the dataset with horizontal flips which constitutes an invariant transformation on the distribution of flower photos.

For evaluation, we calculate the loss (10) and the *Fréchet Inception Distance* (FID) on a held-out test set. The former is a proxy upper bound to the negative log-likelihood, which is associated with the diversity of the generated samples [8, 6]. The latter is a standard metric for image generation tasks and has been observed to capture the "realness" of the generated data distribution [2]. To compute the FID, we first obtain the essential features of both the real and generated images by feeding them through the Inception v3 image classification model and extracting the feature maps after its convolutional block. Then we determine the mean $\boldsymbol{\mu}_{\text{real}}, \boldsymbol{\mu}_{\text{gen}}$ and covariance $\boldsymbol{\Sigma}_{\text{real}}, \boldsymbol{\Sigma}_{\text{gen}}$ over the real and generated image features, respectively, to calculate

$$\text{FID} \coloneqq |\boldsymbol{\mu}_{\text{real}} - \boldsymbol{\mu}_{\text{gen}}| + \operatorname{tr}(\boldsymbol{\Sigma}_{\text{real}} + \boldsymbol{\Sigma}_{\text{gen}} - 2\sqrt{\boldsymbol{\Sigma}_{\text{real}}\boldsymbol{\Sigma}_{\text{gen}}}). \tag{17}$$

We tuned the hyperparameters with respect to these two metrics separately for each of the five noise schedules, and report the best results per schedule in Table 1. The best performance is achieved by the learned schedule, which was to be expected since the ability to dynamically adapt during training is an unfair advantage over fixed schedules. However, the margin is not enormous: in terms of test set loss, the linear schedule is close behind, and on the FID score the cosine schedule is second. The constant and hyperbolic schedules consistently perform worst. Surprisingly, the rather ad-hoc linear schedule performs reasonably well, whereas the carefully crafted squared cosine schedule exhibits a mediocre test set loss.

Table 1: Metrics achieved with different noise schedules.

Schedule	Definition	Test Set Loss	FID Score
constant	(11)	$1.55 \cdot 10^{-4}$	154.8654
linear	(12)	$\mathbf{1.51} \cdot 10^{-4}$	136.7476
hyperbolic	(13)	$2.75 \cdot 10^{-4}$	145.8457
cosine^2	(14)	$2.56 \cdot 10^{-4}$	**131.6321**
learned	(16)	$\mathbf{1.40} \cdot 10^{-4}$	**120.3937**

While the learned schedule appears to be the best overall choice, we would like to highlight the fact that it complicates the implementation considerably and also introduces new hyperparameters for initialisation that require careful tuning. So instead of fully abandoning the hyperparameters of fixed schedules, the learned schedule merely replaces them with other ones.

The caveat of our empirical study is its limited breadth. Future work could include even more schedule choices, train the DPMs on multiple datasets of various sizes, and evaluate the models at different training stages to investigate the dynamics. Due to limited time and computational resources, this was not possible for us.

5 Conclusion

Diffusion probabilistic models are entering the main stage of generative modelling. In this work, we compared several noise schedules both analytically on a conceptual level and empirically on a non-standard dataset. Our best results were achieved with a learned schedule, yet at the cost of more involved implementation and initialisation. Out of the fixed schedules, both a linear and a squared cosine schedule achieve satisfactory performance. However, the impact of the exact schedule choice does not seem to be as significant as one would intuitively expect. Nevertheless, more research needs to be done on this topic to get a definite answer.

References

[1] Prafulla Dhariwal and Alexander Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 8780–8794.

[2] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[4] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". In: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2021.

[5] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations (ICLR*. 2014.

[6] Diederik P. Kingma et al. *Variational Diffusion Models*. 2021.

[7] Calvin Luo. *Understanding Diffusion Models: A Unified Perspective*. 2022.

[8] Alexander Quinn Nichol and Prafulla Dhariwal. "Improved Denoising Diffusion Probabilistic Models". In: *Proceedings of the 38th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, 2021, pp. 8162–8171.

[9] Maria-Elena Nilsback and Andrew Zisserman. "Automated Flower Classification over a Large Number of Classes". In: *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. Image Processing (ICVGIP). Bhubaneswar, India: IEEE, 2008, pp. 722–729.

[10] Aditya Ramesh et al. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. 2022.

[11] Robin Rombach et al. "High-Resolution Image Synthesis With Latent Diffusion Models". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, pp. 10684–10695.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Vol. 9351. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 234–241.

[13] Chitwan Saharia et al. *Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding*. 2022.

[14] Tim Salimans and Jonathan Ho. "Progressive Distillation for Fast Sampling of Diffusion Models". In: *International Conference on Learning Representations*. 2022.

[15] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *Proceedings of the 32nd International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, 2015, pp. 2256–2265.

[16] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *International Conference on Learning Representations*. 2021.

[17] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[18] Daniel Watson et al. *Learning to Efficiently Sample from Diffusion Probabilistic Models*. 2021.